THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

AMA 2233

Data Analytics and Visualization

Yancheng Yuan

Department of Data Science and Artificial Intelligence

Recap of last lecture

Cut of a Graph

Cut of a Graph: Let G = (V, E, W) be a given graph and let (S, S^c) be the partition of the vertices of the graph. Then, the cut of the partition (S, S^c) is defined as

$$Cut(S) = \sum_{i \in S} \sum_{j \in S^c} W_{ij}$$

The cut of a graph is a good measure for a clustering (two clusters) of the vertices of the graph G. In particular, we want to find a partition (S, S^c) of the vertices V which keeps as many of the edges, or connections, within the clusters and has as few edges as possible across clusters.

Compute the cut via Laplacian Matrix

Let G = (V, E, W) be a given graph with n vertices and let (S, S^c) be the partition of the vertices of the graph. Define a vector $y \in R^n$ as

$$y_i = \begin{cases} 1 & \text{if } i \in S, \\ -1 & \text{if } i \in S^c. \end{cases}$$

Then, we have

$$Cut(S) = \frac{1}{4} \sum_{i<j} W_{ij} \left(y_i - y_j\right)^2$$

Compute the cut via Laplacian Matrix

Let G = (V, E, W) be a given graph with n vertices, the Laplacian matrix of G is defined as

$$L_G = D - W,$$

where D is a diagonal matrix with

$$D_{ii} = \sum_{j=1}^{n} W_{ij}.$$

Theorem Let G = (V, E, W) be a given graph with n vertices and L_G is its Laplacian matrix. Let $x \in R^n$ be any given vector, then

$$x^T L_G x = \sum_{i<j} W_{ij}(x_i - x_j)^2$$

Therefore,

$$Cut(S) = \frac{1}{4}\, \mathrm{y}^{\mathrm{T}} \mathrm{L_G y}$$

Model Two Clusters Clustering Problem as a Balanced Minimum Cut

Let G = (V, E, W) be a given graph with n vertices, we can model the clustering of the vertices V of the graph as a balanced minimum cut problem

$$\min_{S \subseteq V, |S| = |S^c|} Cut(S) = \min_{y \in \{-1,1\}^n, 1^T y = 0} \frac{1}{4} y^T L_G y$$

Relaxed balanced minimum cut problem:

$$\min_{||y|| = \sqrt{n}, 1^T y = 0} \frac{1}{4} y^T L_G y$$

If we take $z = y/\sqrt{n}$, we can write the above problem as

$$\min_{||z|| = 1, 1^T z = 0} \frac{n}{4} z^T L_G z$$

Property of the Laplacian Matrix

Theorem Let G = (V, E, W) be a given graph with n vertices, and let L_G be its Laplacian matrix, then

1. All the eigenvalues of L_G are real and nonnegative.
2. 0 is the smallest eigenvalue of L_G.

Remark:

$$L_G = D - W = \sum_{i<j} W_{ij}(e_i - e_j)(e_i - e_j)^T$$

Solution to the Balanced Minimum Cut

Relaxed balanced minimum cut problem:

$$\min_{||y||=\sqrt{n}, 1^T y=0} \frac{1}{4} y^T L_G y$$

If we take $z = y/\sqrt{n}$, we can write the above problem as

$$\min_{||z||=1, 1^T z=0} \frac{n}{4} z^T L_G z$$

Theorem The solution z for the relaxed balanced minimum cut problem is the eigenvector corresponding to the second smallest eigenvalue.

Spectral Clustering for K Clusters

Spectral Clustering Algorithm

Input: A graph G = (V, E, W), number of clusters K.

Step 1: Compute the noralized Laplacian matrix L_G.

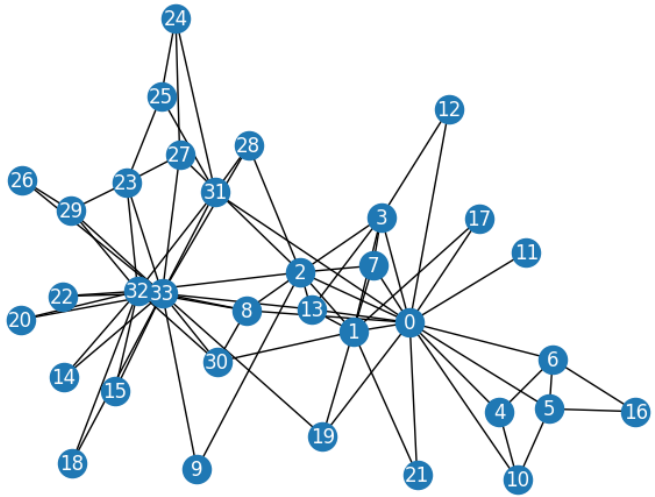
Step 2: Calculate the eigenvectors corresponding to the smallest K eigenvalues of L_G. Form it as $V = [v_1, v_2, \dots, v_K]^T$.

Step 3: Run K-means algorithm on the columns of V.

Applications in Community Detection

Model Two Clusters Clustering Problem as a Balanced Minimum Cut

Community detection in a network identifies and groups the more densely interconnected nodes in a given graph. This graph can take the form of a social network graph, a biological network, or a representation of a local network of computers。



Spectral Clustering for Community Detection

Spectral Clustering Algorithm for Community Detection

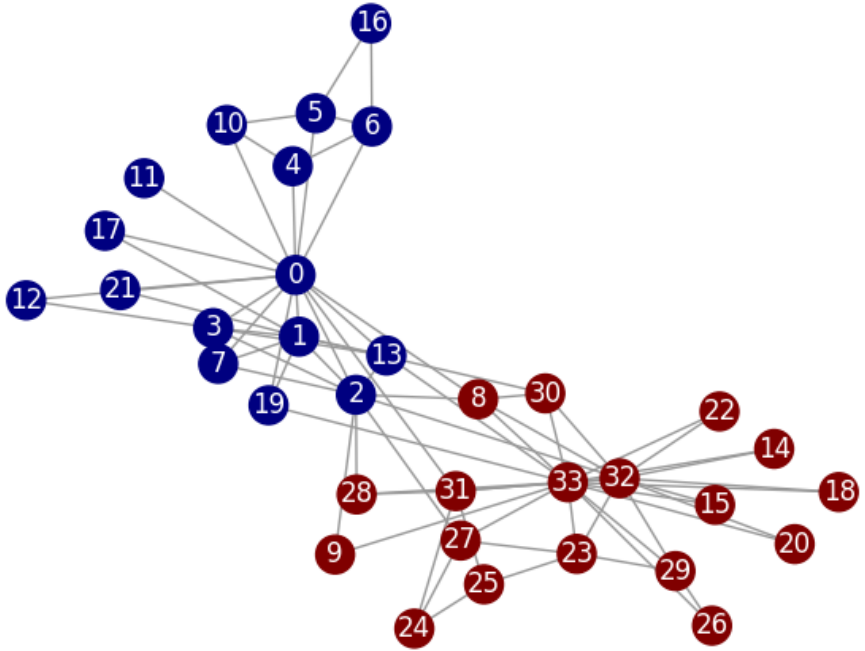
Input: A graph G = (V, E, W), number of clusters K.

Step 1: Compute the noralized Laplacian matrix L_G.

Step 2: Calculate the eigenvectors corresponding to the smallest K eigenvalues of L_G. Form it as $V = [v_1, v_2, \dots, v_K]^T$.

Step 3: Run K-means algorithm on the columns of V.

Application in Karate Club Dataset



Agenda



1. Principal Component Analysis
2. Applications
3. Recap for the Final Exam

Principal Component Analysis

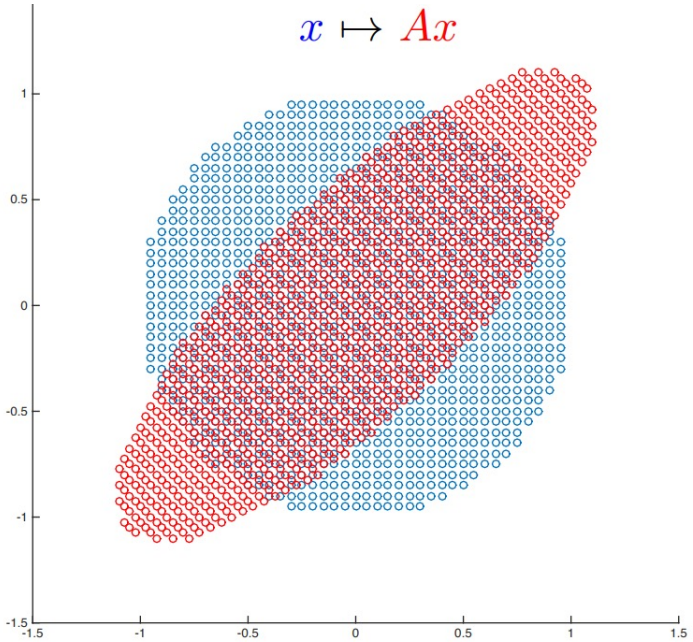
Motivation

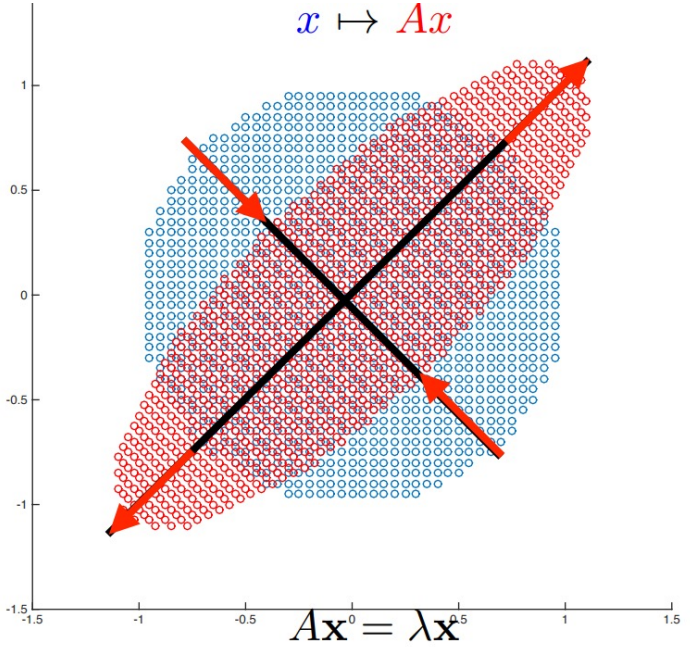
A key motivation for PCA: PCA is important for our course with a focus on visualization. However, we can only plot data with **no more than Dimension 3**. Therefore, a dimension reduction technique for high dimensional data is a must.

Other motivations:

1. **Save computation/memory costs.**
2. **Prevent overfittings.**

Linear Algebra Recap: Eigen Values and Eigenvectors





Linear Algebra Recap: Eigen Decomposition

Eigendecomposition for Symmetric Matrix

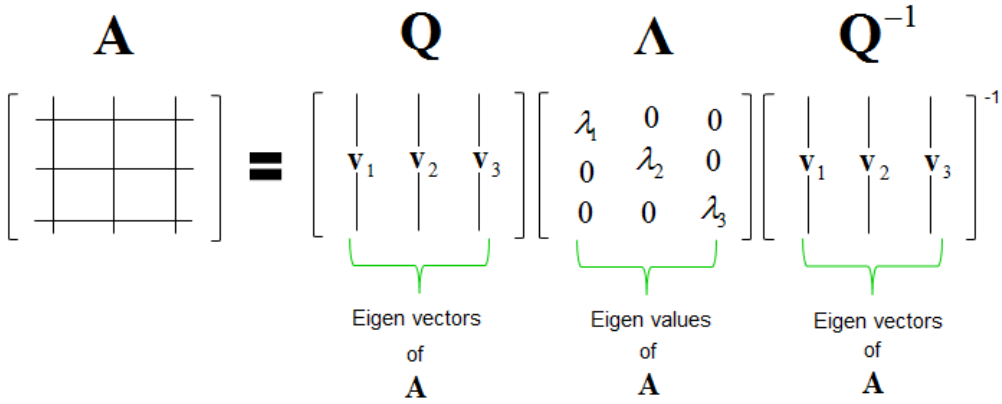
$$A = Q \Lambda Q^{-1}$$
$$= Q \Lambda Q^{T}$$

where $A \in \mathbb{R}^{m \times m}$
↳ Symmetric Square Matrix

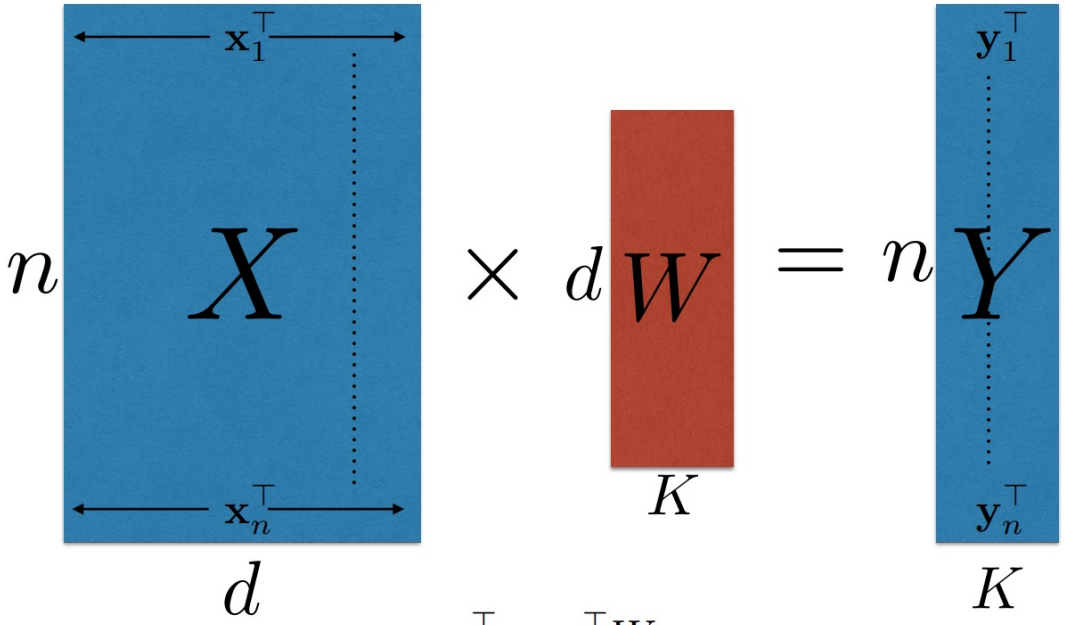
$Q \in \mathbb{R}^{m \times m}$
↳ Orthogonal Matrix

$\Lambda \in \mathbb{R}^{m \times m}$
↳ Diagonal Matrix
(Eigenvalues in the diagonal)

If Q is orthogonal,
$Q^{-1} = Q^{T}$



A demonstration



$$\mathbf{y}_i^\top = \mathbf{x}_i^\top W$$

Goal: Find the matrix W!

Principal Component Analysis: From the Approximation Perspecctive

Idea: Find K orthogonal normal vectors $w_1, \dots, w_K$ such that the projection onto the subspace spanned by these k vectors has a best approximation to the original vectors $x_1, x_2, \dots, x_n$.

$$min_{W,Y,\mu} \frac{1}{2} \sum_{i=1}^{n} ||x_i - \left(\sum_{j=1}^{k} w_j y_{ij} + \mu \right)||^2$$

PCA Theorem

The best k dimensional approximation is given by

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \,,$$

$w_1, w_2, \dots, w_k$ are the first k eigenvectors of the covariance matrix

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T,$$

and

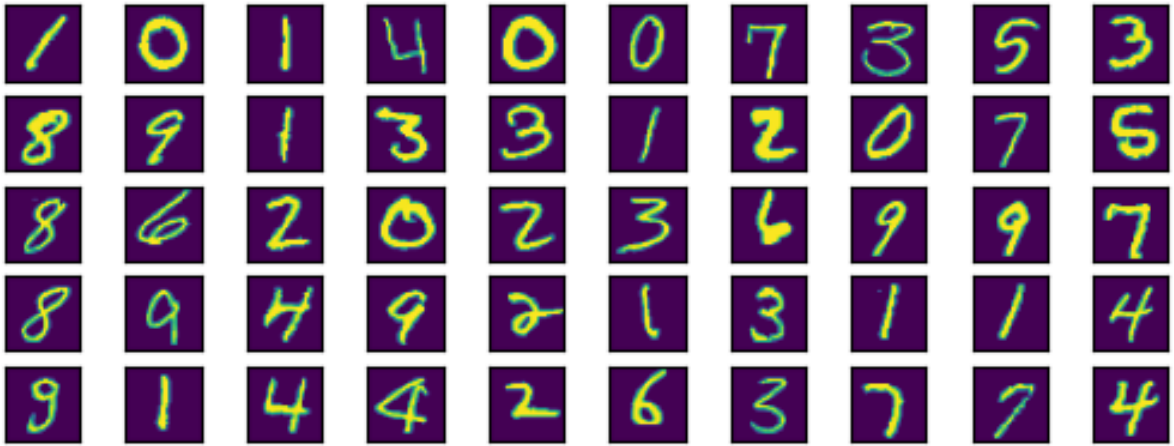
$$y_i^T = (x_i - \mu)^T W$$

1. $\Sigma = \mathrm{cov}\left(X\right)$

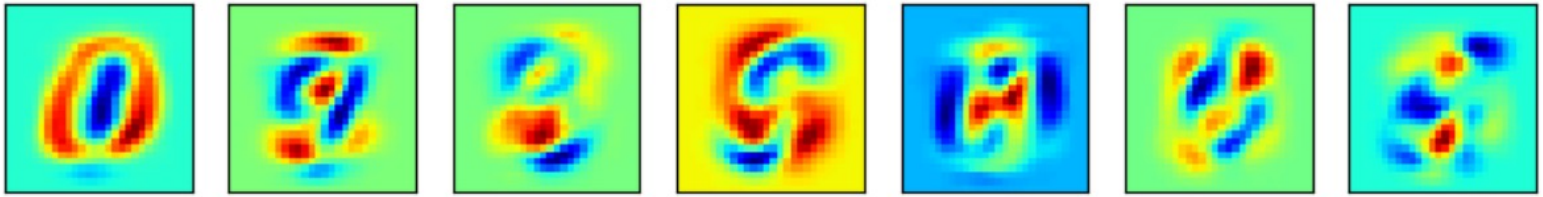
2. $W = \mathrm{eigs}\left(\Sigma, K\right)$

3. $Y = X - \mu \times W$

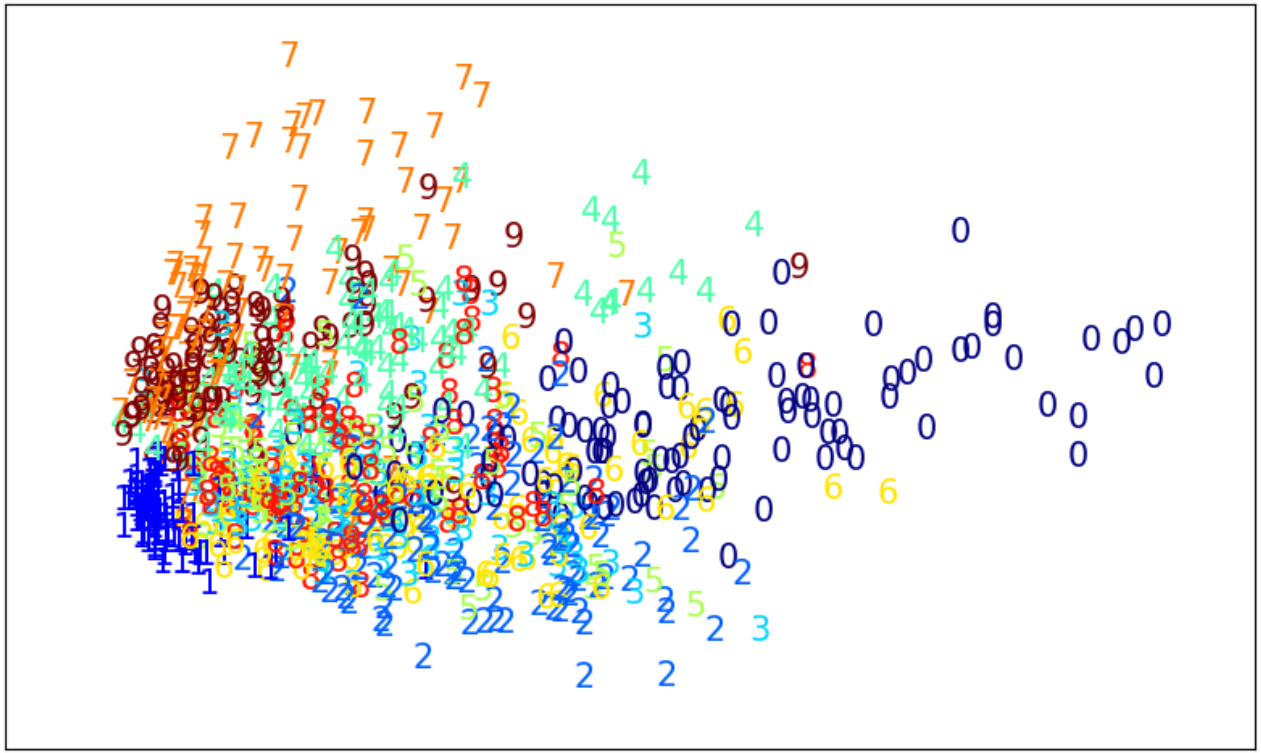
Application in MNIST Data



Visualization of the First 7 Eigenvectors



Visualization in 2D



Recap for Final (handwritten during the lecture)


Thank You!